Escuela
Politécnica
Superior

Reconocimiento de imágenes de flora y fauna

Grado en Ingeniería Informática

Trabajo Fin de Grado

Autor:
Ferran Pérez Esteve
Tutor/es:
Miguel Ángel Cazorla Quevedo
Francisco Gómez Donoso

Mayo 2020

Universitat d'Alacant
Universidad de Alicante

Reconocimiento de imágenes de flora y fauna

Autor
Ferran Pérez Esteve

Tutor/es
Miguel Ángel Cazorla Quevedo
Ciencia de la Computación e Inteligencia Artificial
Francisco Gómez Donoso
Ciencia de la Computación e Inteligencia Artificial

Grado en Ingeniería Informática

Escuela
Politécnica
Superior

Universitat d'Alacant
Universidad de Alicante

ALICANTE, Mayo 2020

Agradecimientos

En este espacio me gustaría darle un pequeño reconocimiento a todas las personas que me han ayudado y han hecho que este trabajo sea posible.

Me gustaría empezar agradeciendo a mis tutores Miguel y Fran la ayuda que me han dado en absolutamente todo lo que he necesitado. Gracias por guiarme y estar pendientes de todos los detalles que han ido surgiendo durante el desarrollo del proyecto. Ha sido un auténtico placer trabajar con vosotros.

También quiero agradecer todo el apoyo que me han proporcionado mi familia y mis amigos, que han estado animándome en todo momento y han sido fundamentales para que este trabajo saliera adelante.

Los ordenadores son inútiles,
solo pueden darnos respuestas

Atribuida a Pablo Picasso

Índice general

Agradecimientos v

Índice de figuras x

Índice de tablas xi

1 Introduccion 1
- 1.1 Descripción del problema . . . 1
- 1.2 Aplicaciones . . . 2
  - 1.2.1 Protección de espacios naturales . . . 2
  - 1.2.2 Control de plagas . . . 2

2 Objetivos y motivación 3

3 Estado del arte 4

4 Metodología 6
- 4.1 Pasos en el desarrollo del proyecto . . . 6
- 4.2 Herramientas utilizadas . . . 7
  - 4.2.1 Software . . . 7
  - 4.2.2 Hardware . . . 7

5 Desarrollo 8
- 5.1 Análisis del conjunto de datos . . . 8
  - 5.1.1 Similitud visual de las imágenes . . . 9
  - 5.1.2 Clases desbalanceadas . . . 10
  - 5.1.3 Calidad de las fotografías . . . 11
  - 5.1.4 Múltiples clases en una imagen . . . 12
- 5.2 Entrenamiento del modelo . . . 13
  - 5.2.1 Modelos utilizados . . . 13
  - 5.2.2 Dimensiones de las imágenes . . . 14
  - 5.2.3 Aumentado de datos . . . 14
  - 5.2.4 Clasificación taxonómica multietiqueta . . . 16
  - 5.2.5 Ensemble learning . . . 17
    - 5.2.5.1 Boosting . . . 17
    - 5.2.5.2 Stacking . . . 18
    - 5.2.5.3 Clasificadores especializados . . . 19
  - 5.2.6 Clasificación por niveles . . . 20

6 Experimentación **22**
6.1 Resultados de los experimentos 22
6.1.1 Primer experimento - Resnet50 22
6.1.2 Aumentado de datos 23
6.1.3 Resolución 24
6.1.4 Arquitecturas 25
6.1.5 Crop 26
6.1.6 Uso de datos de años anteriores 27
6.1.7 Comparación entre entrenamiento y test 28
6.1.8 Resumen de las arquitecturas CNN 29
6.1.9 Ensemble Boosting 30
6.1.10 Clasificación por niveles 31
6.1.11 Clasificación de filos 32
6.2 Clasificación de vídeo 32
6.3 Biodiversidad Virtual 35

7 Conclusiones **38**

Bibliografía **40**

Índice de figuras

5.1 Ejemplo de dos especies con similitud visual 9
5.2 Ejemplo de imagen con el elemento reducido 10
5.3 Gráfica con el número de elementos por clase 10
5.4 Fotografías que muestran las diferentes calidades de los datos 11
5.5 Fotografías en las que el elemento se ve parcialmente 12
5.6 Fotografía donde aparece más de un elemento 12
5.7 Misma imagen con diferente resolución 14
5.8 Aumentado de datos con crop 16
5.9 Ejemplo de clasificación multietiqueta 17
5.10 Arquitectura stacked 18
5.11 Arquitectura con modelos especializados 19
5.12 Clasificación de niveles 21

6.1 Acierto en el género Bombus de Biodiversidad 22
6.2 Resultados de utilizar data augmentation 23
6.3 Experimento resolución 1 25
6.4 Experimento resolución 2 25
6.5 Acierto Top 1 - Experimento con diferentes arquitecturas 26
6.6 Acierto Top 5 - Experimento con diferentes arquitecturas 26
6.7 Acierto - Experimento con crop 27
6.8 Acierto - Experimento con datos de otros años 28
6.9 Comparación entre entrenamiento y test 29
6.10 Clasificación en vídeo de *Bombus vosnesenskii* 33
6.11 Clasificación en vídeo de *Bombus terrestis* 33
6.12 Clasificación en vídeo de *Lupinus texensis* 34
6.13 Clasificación en vídeo de *Thamnophis sirtalis* 34
6.14 Especies de culebras similares 35
6.15 Clasificación en vídeo de *Tringa flavipes* 35
6.16 Clasificación en vídeo de *Setophaga petechia* 36
6.17 Imágenes de Bombus de Biodiversidad Virtual 36
6.18 Acierto en el género Bombus de Biodiversidad 37

Índice de tablas

3.1 Ejemplos de conjuntos de datos de grano fino 5

5.1 Árbol taxonómico 8
5.2 Principales grupos del conjunto de datos 9
5.3 Cantidad de ejemplos para cada subconjunto 13
5.4 Modelos utilizados para la clasificación de flora y fauna 13
5.5 Ejemplo de aumentado de datos 15

6.1 Parámetros en el primer experimento 23
6.2 Parámetros usados en el experimento de data augmentation 23
6.3 Parámetros usados en el experimento 1 de resolución 24
6.4 Parámetros usados en el experimento 2 de resolución 24
6.5 Parámetros usados en el experimento 2 de resolución 25
6.6 Parámetros usados en el experimento del crop 27
6.7 Parámetros usados en el experimento con datos de años anteriores 27
6.8 Resultados de las categorías taxonómicas 28
6.9 Tabla resumen con los resultados de las diferentes arquitecturas 29
6.10 Párametros Ensemble 1 30
6.11 Párametros Ensemble 2 30
6.12 Párametros Ensemble 3 30
6.13 Resultados Ensemble 31
6.14 Párametros clasificación por niveles 31
6.15 Resultados clasificación por niveles 31
6.16 Párametros clasificación por filos 32
6.17 Resultados clasificación por filos 32
6.18 Género Bombus en Biodiversidad Virtual 36

1 Introduccion

1.1 Descripción del problema

Se estima que el mundo natural contiene millones de especies de plantas y animales. Sin conocimiento experto es extremadamente complicado clasificar muchas de estas especies debido a su gran similitud visual. El objetivo de este trabajo es explorar el reconocimiento automático de imágenes, con la finalidad de desarrollar un clasificador robusto que sea capaz de diferenciar entre más de mil clases de seres vivos con características similares. En este trabajo, se hará uso de redes neuronales convolucionales para la construcción del modelo. Esta clase de arquitectura es ampliamente usada en la actualidad para tareas de clasificación de imágenes.

Este problema se enmarca dentro del conocido como *Fine-Grained Visual Categorization*, consistente en la clasificación de imágenes que comparten un gran número de características similares entre sí. Es un tema que ha recibido una atención significativa recientemente debido a los avances en Deep Learning y en el reconocimiento de imágenes. El principal interés detrás de este problema es capturar las sutiles diferencias visuales entre las diferentes categorías.

El conjunto de datos con el que contamos para entrenar y validar el modelo proviene del *iNaturalist Challenge 2019*, que contiene imágenes de 1010 categorías diferentes. La variedad de especies a clasificar es muy alta, incluyendo entre otras, imágenes de insectos, reptiles, anfibios, pájaros y todo tipo de plantas. Las fotografías han sido tomadas en una gran variedad de situaciones y provienen de lugares de todo el mundo. El modelo por tanto, debe ser capaz de adaptarse a estas condiciones, teniendo que clasificar correctamente bajo un amplio abanico de escenarios, donde la distancia, la luz y el fondo, entre otros factores, pueden cambiar significativamente en cada imagen. La gran variedad de situaciones que contienen los datos, junto al amplio número de especies a clasificar hacen que este problema sea especialmente desafiante.

El clasificador, por tanto, recibirá una única imagen que contendrá algún elemento de flora o fauna perteneciente a las categorías incluidas en el conjunto de datos, y deberá predecir correctamente de qué especie concreta se trata. Además, también podrá reconocer a qué categorías taxónomicas pertenece el elemento, dando como resultado el árbol taxónomico completo. Por todos los factores que hemos comentado, se trata de una tarea que solamente un experto en la materia podría realizar, por lo que desarrollar una solución efectiva puede tener múltiples aplicaciones.

1.2 Aplicaciones

1.2.1 Protección de espacios naturales

La resolución de este problema cobra especial importancia en el mundo actual, donde cada día aumenta el número de especies en peligro de extinción. Miles de animales y plantas están amenazados por diversos motivos, principalmente, debido a la destrucción de sus hábitat y al cambio climático. Por ello, contar con un modelo capaz de reconocer imágenes de flora y fauna puede ayudar a la prevención y protección de estas especies, ya que proporciona una forma sencilla y accesible de identificarlas.

La utilización de este sistema podría tener una aplicación directa en los espacios naturales protegidos, donde se suele tener un alto control de las especies que viven en él, especialmente si estas se encuentran en peligro de extinción. Empleando un sistema de cámaras, se podrían localizar fácilmente los distintos especímenes, haciendo de su seguimiento y protección una tarea más sencilla.

Los visitantes de estos espacios naturales sin conocimientos expertos también podrían sacar provecho de este sistema, ya que contarían con una herramienta que les permitiría identificar las especies vulnerables de la zona y, además, podrían obtener información muy valiosa sobre la biodiversidad del territorio. Con este sistema no solo podrían identificar la especie, sino que también podrían conocer su taxonomía, aportando información que permitiría a los usuarios aumentar sus conocimientos sobre el mundo natural.

1.2.2 Control de plagas

El reconocimiento automático de imágenes en especies animales y vegetales puede ser de gran ayuda en el mundo de la agricultura, especialmente en la tarea de identificar especies que pueden causar daños en los cultivos. El sistema podría permitir la identificación visual de estas especies de forma práctica y rápida, contando únicamente con una fotografía. Obviamente, para realizar esta tarea se necesitaría contar previamente con un conjunto de datos etiquetados, donde estuvieran todas las especies invasoras que se quisieran identificar.

Típicamente este es un problema que requeriría conocimientos expertos, pero con un sistema de cámaras repartidas de forma estratégica, se podría facilitar esta tarea, detectando automáticamente si en alguna zona ha aparecido una especie que pueda causar daños en los cultivos. De esta forma, se podrían prevenir y neutralizar estas plagas de forma efectiva.

2 Objetivos y motivación

La motivación principal para desarrollar este proyecto fue el interés que tenía en aprender y aplicar tecnologías relacionadas con deep learning en una situación real. Además, la complejidad del problema me ha permitido experimentar con diversas técnicas y arquitecturas para la clasificación de imágenes. Otra de las razones que más me atrajeron fue la relación directa que tenía el proyecto con el mundo natural, y su gran utilidad para catalogar y preservar la biodiversidad natural, tanto en especies de fauna como de flora.

Objetivos a conseguir durante el desarrollo del proyecto:

- Realizar un análisis de los problemas que conlleva el reconocimiento de imágenes en especies naturales similares entre sí, evaluando el conjunto de datos y sus características
- Implementar un clasificador basado en redes neuronales convolucionales que sea capaz de reconocer imágenes de flora y fauna, tratando que este tenga el mayor porcentaje posible.
- Hacer uso de las distintas tecnologías y librerías que se usan actualmente en la implementación de sistemas de machine learning, entre las que se incluyen Pandas, Keras, TensorFlow y Scikit-learn.
- Participar en el desafío de Kaggle *iNaturalist 2019 at FGVC6*, con el propósito de validar el modelo construido y compararlo con otras soluciones existentes.
- Construir una solución que tenga en cuenta la taxonomía de cada especie, tanto en el entrenamiento del modelo como en la clasificación de las imágenes.
- Realizar una comparación experimental entre las diferentes soluciones que se han probado, con el objetivo de analizar las ventajas y los inconvenientes de cada una.
- Explicar de forma detallada el funcionamiento de las distintas soluciones, teniendo en cuenta los parámetros que se han utilizado.

3 Estado del arte

La clasificación multiclase de imágenes tiene un largo recorrido en la historia de la inteligencia artificial. Uno de los enfoques más simples para realizar la clasificación multiclase es entrenar una serie de clasificadores binarios, donde la salida de cada clasificador se utiliza para producir la clasificación multiclase final. Este enfoque se exploró en redes neuronales tempranas y en las máquinas de vectores de soporte. La solución, aunque es simple, también tiene serios inconvenientes. Por ejemplo, el espacio de características no se explora correctamente y puede generar problemas de sobreajuste.

En el pasado se introdujo teóricamente un clasificador único que realizara predicciones multiclase, no obstante, no se pudo probar debido a la falta de potencia de cálculo. Sin embargo, con la emergencia de las plataformas masivamente paralelas como las GPU, este enfoque ha sido ampliamente explorado. En [1], [2] y [3], se compara el enfoque tradicional de un conjunto de clasificadores binarios con un clasificador multiclase puro. Los autores afirman que el enfoque multiclase tiene varias ventajas, como la reducción del tiempo de capacitación e inferencia y una exploración más amplia del espacio de características. Sin embargo, estos enfoques no aprovechan que las etiquetas se organizan de manera taxonómica. En el mundo real, casi todas las categorías podrían dividirse en una variedad de varias subcategorías con ciertas características en común. Por ejemplo, la categoría de vehículos podría dividirse en automóviles, motocicletas, camiones y furgonetas.

En [4], se concluye que la clasificación multiclase plana funciona mejor con datos bien balanceados, mientras que el enfoque que utiliza un conjunto de clasificadores funciona mejor en presencia de datos no balanceados. Por lo tanto, es preferible el uso de varios clasificadores en conjuntos de datos como Inaturalist, donde el número de ejemplos que contiene cada especie está altamente desbalanceada.

El problema que queremos resolver pertenece a un campo específico de la clasificación de imagen, en el que las distintas categorías tienen grandes similitudes visuales, este tipo de identificación se denomina habitualmente como clasificación de grano fino. Existen dos diferencias distintivas respecto a la clasificación de imágenes habitual. Primero, tiende a haber solo un pequeño número de expertos que son capaces de hacer las clasificaciones. En segundo lugar, a medida que avanzamos por el espectro de la similitud, el número de instancias en cada clase se vuelve menor. Esto motiva la necesidad de sistemas automatizados que sean capaces de discriminar entre un gran número de categorías potencialmente similares, contando únicamente con un pequeño número de ejemplos para algunas categorías.

Como ejemplo, la identificación facial puede verse como un tipo de clasificación con características visuales similares entre sí. Sin embargo, debido a la similitud geométrica subyacente

entre caras, los enfoques actuales para la identificación de caras tienden a realizar una gran cantidad de preprocesamiento específico de caras [5], [6], [7].

Los conjuntos de datos de grano fino también suelen tener otro problema, el número de imágenes suele ser menor, ya que es más difícil obtener anotaciones detalladas de expertos. Para solucionar este problema, [8] propuso un esquema de aprendizaje de transferencia de datos. Sin embargo, esta técnica requiere reentrenar modelos usando grandes conjuntos de datos sin obtener un rendimiento significativo en comparación con el tiempo de cómputo. Actualmente, existen conjuntos de datos de grano fino que cubren varios dominios, como aves [9], [10], [11] y perros [12] , [13]. El conjunto ImageNet [14] no se suele considerar como un conjunto de datos de grano fino, no obstante, contiene varios grupos de clases de grano fino, incluidas alrededor de 60 especies de aves y de 120 razas de perros. Muchos de estos conjuntos de datos se construyeron con la intención de que tuvieran un distribución uniforme de imágenes en las diferentes categorías.

Las imágenes de especies naturales tienden a ser desafiantes ya que los individuos de la misma especie pueden diferir en apariencia, debido al sexo y la edad. Además, también pueden aparecer en diferentes ambientes. Dependiendo de la especie en particular, pueden ser muy difíciles de fotografiar en la naturaleza. Por el contrario, las categorías de objetos hechos por el hombre solamente suelen diferir en términos de pose, iluminación o color. Pero no necesariamente en la forma o apariencia del objeto subyacente.

Conjuntos de datos populares de grano fino			
Nombre	Ejemplos	Categorias	Desbalanceo
Oxford Pets [13]	3,680	37	1.08
CUB 200-2011 [9]	5,994	200	1.03
ILSVRC2012 [14]	1,281,167	1,000	1.78
INaturalist 2019	268,243	1,010	31.25

Tabla 3.1: Ejemplos de conjuntos de datos de grano fino

En [15] se realiza un análisis muy interesante sobre el conjunto de iNaturalist 2017, comparándolo con otros conjuntos de datos popular de grano fino. Utilizando como medida de desbalanceo, el número de imágenes en la clase más representada dividida por la que tiene menos ejemplos. A pesar de los valores atípicos, nos da una indicación del desbalanceo presente en iNaturalist. En la Tabla 3.1 aparecen estos datos actualizados a INaturalist 2019. Por todo lo que hemos comentado, podemos decir que este conjunto de datos es realmente desafiante para los modelos y soluciones actuales, debido al desbalanceo de datos y a las grandes similitudes visuales.

4 Metodología

En este capítulo explicaremos cuál ha sido el proceso de desarrollo y análisis, así como las herramientas de software y de hardware que se han utilizado para la implementación.

4.1 Pasos en el desarrollo del proyecto

1. **Estudio de las redes CNN y de las diferentes arquitecturas de clasificación multiclase**

   El primer paso ha sido realizar un estudio de las técnicas que se usan actualmente en la clasificación de imágenes multiclase, poniendo especial atención en aquellos métodos enfocados en identificar imágenes similares entre sí con un conjunto de datos desbalanceados.

2. **Análisis del conjunto de datos**

   Uno de los puntos claves en el aprendizaje supervisado son los datos usados para el entrenamiento. Por ello, se ha llevado a cabo un análisis exhaustivo de iNaturalist 2019, que contiene las imágenes que se han usado para desarrollar el modelo. Es importante detectar cuáles son los potenciales problemas e inconvenientes que tiene este conjunto de datos para su posterior uso.

3. **Implementación del procesado de datos y de la arquitectura**

   Una vez realizada la parte más analítica, se debe proceder a la implementación del modelo. En esta parte se incluyen las técnicas utilizadas para procesar y acceder al conjunto de datos de forma eficiente. También se deben implementar todas las arquitecturas que posteriormente se utilizarán y validarán.

4. **Entrenamiento y experimentación**

   En esta fase se han probado las diferentes arquitecturas y técnicas, con el objetivo de compararlas y obtener el máximo rendimiento. Los experimentos se han realizado cambiando múltiples parámetros, entre los que se incluyen diversas técnicas de predicción, transformaciones en las imágenes y varias arquitecturas de deep learning.

4.2 Herramientas utilizadas

4.2.1 Software

En este punto vamos a describir brevemente el software que se ha utilizado para el desarrollo del proyecto:

TensorFlow[1]: TensorFlow es una plataforma de código abierto enfocada a la implementación de sistemas de aprendizaje automático, capaces de detectar y descifrar patrones y correlaciones, análogos al aprendizaje y razonamiento usados por los humanos. Se puede utilizar junto una API de alto nivel como Keras.

Keras[2]: Es una API de redes neuronales de alto nivel, escrita en Python y capaz de ejecutarse sobre TensorFlow, CNTK y Theano. Fue desarrollada con el objetivo de permitir la experimentación rápida en redes convolucionales y redes recurrentes. Es capaz de ejecutarse tanto en CPU como en GPU.

HDF5[3]: API enfocada al tratamiento de datos con un alto rendimiento y eficiencia. Permite administrar, procesar y almacenar datos heterogéneos. En nuestro caso se utiliza para almacenar las imágenes y sus etiquetas, nos permite acceder a los archivos de forma eficiente y disminuir el tiempo de entrenamiento.

4.2.2 Hardware

Para el desarrollo de sistemas de aprendizaje con redes neuronales, es necesario contar con GPUs de alto rendimiento que nos permitan ejecutar cálculos en paralelo y obtener unos tiempos de entrenamiento admisibles, para ello, se ha utilizado el siguiente equipo:

Servidor Jackson

- SO: Ubuntu 16.04
- CPU: Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz
- RAM: 16GB DDR4
- GPU 1: NVIDIA QUADRO P6600
- GPU 2 NVIDIA RTX 2080ti

[1] https://www.tensorflow.org/
[2] https://keras.io/
[3] https://www.hdfgroup.org/

5 Desarrollo

En este apartado se explicará paso a paso y de forma detallada todo aquello que se haya implementado y desarrollado a lo largo del proyecto.

5.1 Análisis del conjunto de datos

Los datos con los que contamos para entrenar y validar el modelo están compuestos por un total de 268.243 imágenes, cada una de ellas contienen alguna de las diferentes especies animales y vegetales a clasificar. También tenemos 35.351 imágenes sin ningún tipo de etiqueta, que son las que hay que clasificar correctamente para el desafío de Kaggle.

Categorías taxonomicas	
Nombre	Número de clases
Reino	3
Filo	4
Clase	9
Orden	34
Familia	57
Género	72
Especie	1010

Tabla 5.1: Árbol taxonómico

Todas las imágenes están etiquetadas con la especie a la que pertenece cada individuo. Además, para cada caso contamos con el árbol taxonómico completo de la especie correspondiente. En la Tabla 5.1 se especifica la forma de este árbol. El objetivo principal es predecir correctamente el último nivel del árbol, no obstante, la estructura de los datos nos concede una valiosa información que se puede utilizar para mejorar el rendimiento del modelo.

Las imágenes se dividen en seis grandes grupos diferentes, tal y como podemos ver en la Figura 5.2. Entre estos grupos encontramos diferencias visuales bastante importantes. Los diversos subgrupos en los que se va dividiendo el conjunto tiene cada vez una similitud mayor.

El conjunto de datos presenta una serie de características y dificultades que pueden causar un impacto en el rendimiento cuando entrenemos y validemos al modelo. Es por ello, que se deben analizar cuáles son estos problemas con el objetivo de intentar solventarlos.

(a) Reptil

(b) Anfibio

(c) Pájaro

(d) Insecto

(e) Hongo

(f) Planta

Tabla 5.2: Principales grupos del conjunto de datos

5.1.1 Similitud visual de las imágenes

Como ya se ha comentado en apartados anteriores, uno de los mayores desafíos que presenta este problema es la gran similitud que presentan las clases entre sí. Las especies que comparten las mismas categorías taxonómicas se parecen más entre sí, y en algunos casos, solamente se pueden distinguir a partir de algún pequeño detalle.

(a) Especie de abeja 1

(b) Especie de abeja 2

Figura 5.1: Ejemplo de dos especies con similitud visual

Tomemos como ejemplo la Figura 5.1. En este caso nos encontramos con dos especies diferentes de abejas, pero como se puede observar se parecen mucho entre sí, solamente se distinguen en algunas características sutiles en los colores o en la forma del animal.

El hecho de que necesitemos pequeñas diferencias para clasificar correctamente encierra un

problema, puede que en la imagen no se aprecien estas características. La gran variedad de situaciones en los datos, con imágenes en las que el elemento a clasificar es muy pequeño o solamente se ve parcialmente, provoca que estas diferencias sean prácticamente inapreciables.

Figura 5.2: Ejemplo de imagen con el elemento reducido

Volviendo al ejemplo anterior, la Figura 5.2 contiene una abeja con unas dimensiones bastante reducidas. Identificarla correctamente en estos casos, entraña una complejidad mucho mayor que en las imágenes en las que el elemento a clasificar aparece de forma clara. Ya que no se trata solamente de identificar que en la imagen aparece una abeja, sino a qué especie concreta pertenece.

5.1.2 Clases desbalanceadas

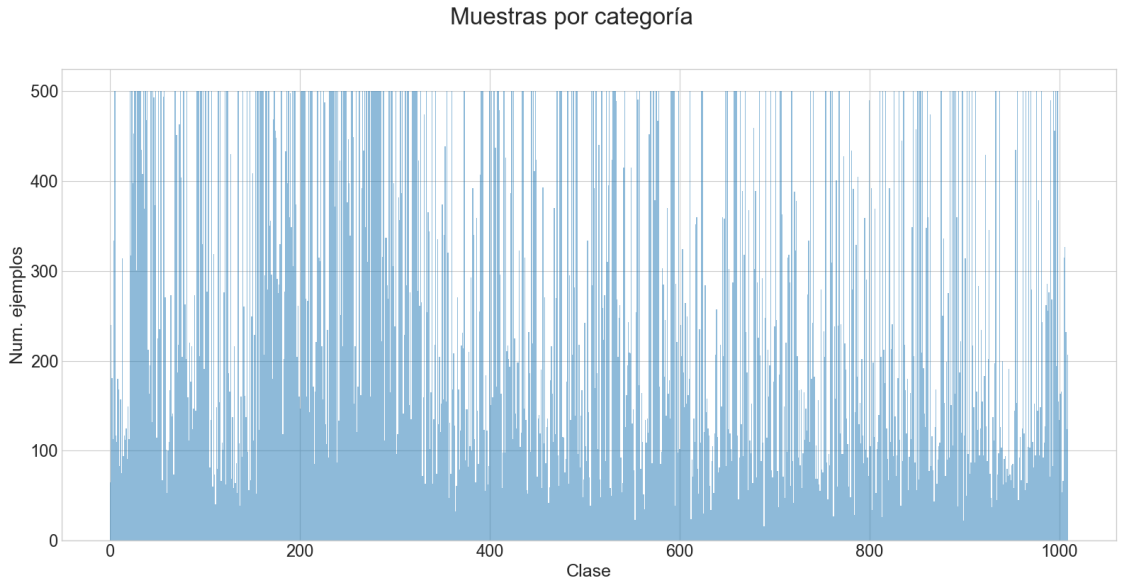


Figura 5.3: Gráfica con el número de elementos por clase

El conjunto de datos está desbalanceado, por lo que la cantidad de ejemplos con los que contamos de clase difiere entre sí. Típicamente, cuando reducimos el número de ejemplos de una clase, el rendimiento del modelo sufre para clasificarla de forma correcta.

En la Figura 5.3 aparecen el número de ejemplos que tenemos de cada clase en el conjunto de datos. Como se puede observar, hay un claro desbalanceo en la cantidad de imágenes que tenemos en cada categoría. Mientras las clases más representadas cuentan con 500 ejemplos, la más infrarepresentada solamente tiene 16 imágenes.

5.1.3 Calidad de las fotografías

Todas las fotografías del conjunto de datos han sido tomadas por usuarios de iNaturalist, usando cámaras con diferente calidad. Por lo tanto, contaremos imágenes de todo tipo:

(a) Imagen de alta resolución **(b)** Imagen borrosa

Figura 5.4: Fotografías que muestran las diferentes calidades de los datos

Fotografías de alta calidad: Gran parte de las imágenes con las que contamos tienen una alta resolución y el elemento a clasificar aparece claramente. La Figura 5.4(a) refleja un caso en el que la especie a identificar se muestra visiblemente y con una buena calidad.

Fotografías borrosas o de baja calidad: En algunas ocasiones tanto el sujeto a identificar como el propio fondo de la imagen aparecen de forma borrosa o con una calidad muy baja. Estos casos, obviamente, serán más difíciles de predecir que aquellos en la que la calidad sea mayor. Un ejemplo de esto, lo podemos ver en la Figura 5.4(b), donde la imagen tiene una baja resolución y tanto el pájaro como el fondo se presentan de forma borrosa.

Fotografías en las que el elemento se ve parcialmente: Hay casos, en los que independientemente de la calidad de la imagen, el sujeto a identificar apenas es perceptible o se encuentra bastante escondido en la imagen. En la Figura 5.5a podemos ver que en la fotografía aparece un anfibio, pero solamente se ve una parte de él, ya que está oculto entre la hierba. El caso de la Figura 5.5b es aún mas extremo, ya que en este ejemplo, la imagen contiene un lagarto (se encuentra en la zona marcada en rojo), pero está tan lejos y escondido entre la maleza, que prácticamente su visibilidad es nula.

(a) El sujeto se ve parcialmente **(b)** El sujeto apenas es perceptible

Figura 5.5: Fotografías en las que el elemento se ve parcialmente

Estas características en los datos condicionarán en gran medida el funcionamiento de nuestro clasificador, ya que los problemas que tengan las imágenes se trasladarán al modelo. El sistema aprende a partir de los datos que nosotros les suministremos, y si le entrenamos con fotografías borrosos o donde el elemento ni siquiera se ve, el rendimiento será menor.

5.1.4 Múltiples clases en una imagen

(a) Imagen etiquetada como fauna **(b)** Imagen etiquetada como flora

Figura 5.6: Fotografía donde aparece más de un elemento

Uno de los problemas de identificar al mismo tiempo flora y fauna es que en algunas fotografías aparecen al mismo tiempo elementos pertenecientes a diferentes clases. Es decir, en una imagen etiquetada como vegetación pueden aparecer insectos u otros animales. También se da con mucha frecuencia el caso contrario, que en un ejemplo etiquetado como fauna aparezca mayoritariamente vegetación.

En la Figura 5.6b aparece una abeja con unas dimensiones reducidas, pero la mayor parte de la imagen está ocupada por una flor. Como en el conjunto de datos hay varios tipos de flores, es posible que en este caso identificara la imagen con algún tipo de vegetación. En la Figura 5.6b sucede el caso inverso, la fotografía contiene un insecto pero la categoría etiquetada es un tipo de planta.

5.2 Entrenamiento del modelo

Una vez analizado el conjunto de datos, el siguiente paso es explicar cómo se ha entrenado el modelo, teniendo en cuenta los diferentes parámetros y técnicas que se han utilizado para realizarlo. Lo primero es dividir los datos en dos subjuntos, uno servirá para el entrenamiento y otro para la validación.

Conjunto de datos	
Entrenamiento	Test
187.770	80.529

Tabla 5.3: Cantidad de ejemplos para cada subconjunto

En la Tabla 5.3 vemos la cantidad de ejemplos que tiene cada subconjunto, en nuestro caso utilizamos un 70% de los datos para el entrenamiento, y 30% para el test. Todas las clases están representadas de forma proporcional en cada subconjunto.

5.2.1 Modelos utilizados

Arquitecturas usadas	
Nombre	Número de parámetros
Resnet50	25,636,712
InceptionV3	23,851,784
Densenet201	19,632,106
EfficientNetB3	12,578,152
EfficientNetB5	19,862,340

Tabla 5.4: Modelos utilizados para la clasificación de flora y fauna

Para resolver nuestro problema, se han usado arquitecturas de redes neuronales convolucionales ya existentes. El objetivo es entrenarlos con los datos de nuestro problema y verificar el rendimiento de cada uno de ellos. La Tabla 5.4 contiene los distintos modelos que se han probado junto al número de parámetros que tiene cada uno.

Estos modelos han sido previamente entrenados con ImageNet, un conjunto de datos que contiene más de 14 millones de imágenes y más de 20.000 categorías diferentes. Actualmente, ImageNet se usa como referente en el reconocimiento de imágenes. Por lo tanto, los pesos de las diferentes arquitecturas se inicializan con los mismos que la red preentrenada. En una red neuronal convolucional, las capas van aprendiendo distintos niveles de abstracción, siendo las primeras capas las encargadas de aprender características más genéricas. El objetivo de utilizar modelos preentrenados es utilizar estas primeras capas, con el objetivo de aplicar las características aprendidas a otros problemas.

5.2.2 Dimensiones de las imágenes

Como hemos comentado en el análisis del conjunto de datos, las imágenes tienen distintas resoluciones, teniendo como máximo un tamaño de 800x800 píxeles y como mínimo, 500x500 píxeles.

La red espera que todas las imágenes tengan el mismo tamaño, por lo que todos los datos se deben reescalar para que tengan las mismas dimensiones. Por otro lado, utilizar una resolución muy alta implica que el tiempo de computación y la cantidad de memoria necesaria es prohibitiva. Es por ello, que se debe utilizar un tamaño menor que los 500x500 píxeles, aunque ello implique una pérdida en el detalle de las fotografías.

En nuestro caso, se han elegidos tamaños de 250x250 y de 300x300 píxeles con tres canales RGB para entrenar nuestros modelos. En la Figura 5.7 vemos un ejemplo de la pérdida de calidad que conlleva este reescalado en la imagen. Como veremos en apartados posteriores, la calidad de la imagen es muy importante para obtener un alto rendimiento en este problema, ya que los pequeños detalles que se pierden pueden ser clave para realizar la clasificación correctamente.

(a) Imagen original **(b)** Imagen reescalada a 300x300 px

Figura 5.7: Misma imagen con diferente resolución

5.2.3 Aumentado de datos

El aumentado de datos, es una técnica que tiene como objetivo incrementar los ejemplos que tenemos para realizar el entrenamiento, utilizando como base los datos ya existentes. En el campo de las imágenes, se realizan transformaciones como rotaciones, recortes, cambios en el brillo y el contraste, o se le añade ruido entre otras muchas técnicas. En nuestro caso, el uso del aumentado de datos tiene como objetivo principal equilibrar las clases con menos representación, creando de forma artificial un balanceo en el conjunto de datos. Por lo tanto, lo que se propone, es crear para cada categoría tantos datos aumentados como sean necesarios para igualarla a la clase más representada. Para el entrenamiento se han utilizado esencialmente dos técnicas de aumentado de datos:

Aumentado de datos sin crop: En este caso, en cada imagen se modifican cierto parámetros de forma aleatoria, aunque dentro de unos rangos establecidos. En la Tabla 5.5 vemos los resultados del aumentado de datos, aplicado a una imagen de planta y a otra de una ave.

Las transformaciones que se aplican son las siguientes:

- Volteado horizontal y vertical
- Desplazamiento en el canal de color.
- Ruido gaussiano
- Cambio en el contraste de la imagen
- Cambio en el brillo de la imagen
- Desenfoque gaussiano

Tabla 5.5: Ejemplo de aumentado de datos

Aumentado de datos con crop: Este tipo de aumentado de datos aplica las mismas transformaciones que en la técnica anterior, pero añade un recorte sobre la imagen original. La intención de este recorte, es que el modelo tenga que aprender a etiquetar imágenes con diferentes resoluciones. En la mayoría de fotografías, el elemento a clasificar se encuentra en el centro, por lo que es de esperar que el recorte no elimine la información necesaria para identificar al sujeto. Si además vamos reduciendo en cada época el recorte que le hacemos a las imágenes, estaríamos aplicando un crop piramidal.

(a) Imagen original **(b)** Imagen con crop

Figura 5.8: Aumentado de datos con crop

En la Figura 5.8 vemos un ejemplo de este tipo de aumentado datos. Como se puede observar, en la imagen aumentada el elemento a identificar, en este caso una flor, aparece con un mayor tamaño que en la imagen original debido al recorte. De igual forma que en la técnica anterior, se le siguen aplicando transformaciones a la imagen.

5.2.4 Clasificación taxonómica multietiqueta

Como se ha comentado en apartados anteriores, cada imagen tiene asociadas múltiples etiquetas, cada una de ellas se corresponde con un categoría taxonómica diferente. Lo que se propone es construir un modelo multietiqueta que tenga que predecir al árbol taxonómico al completo. El objetivo de este método es combinar la predicción de cada etiqueta, con la intención de mejorar el acierto en el último nivel.

En la Figura 5.9 se puede ver el esquema que sigue esta tipo de etiquetado, en el ejemplo, vemos una imagen de una libélula, el modelo tiene que predecir todas las categorías taxónomicas a las que pertenece, incluyendo obviamente la especie.

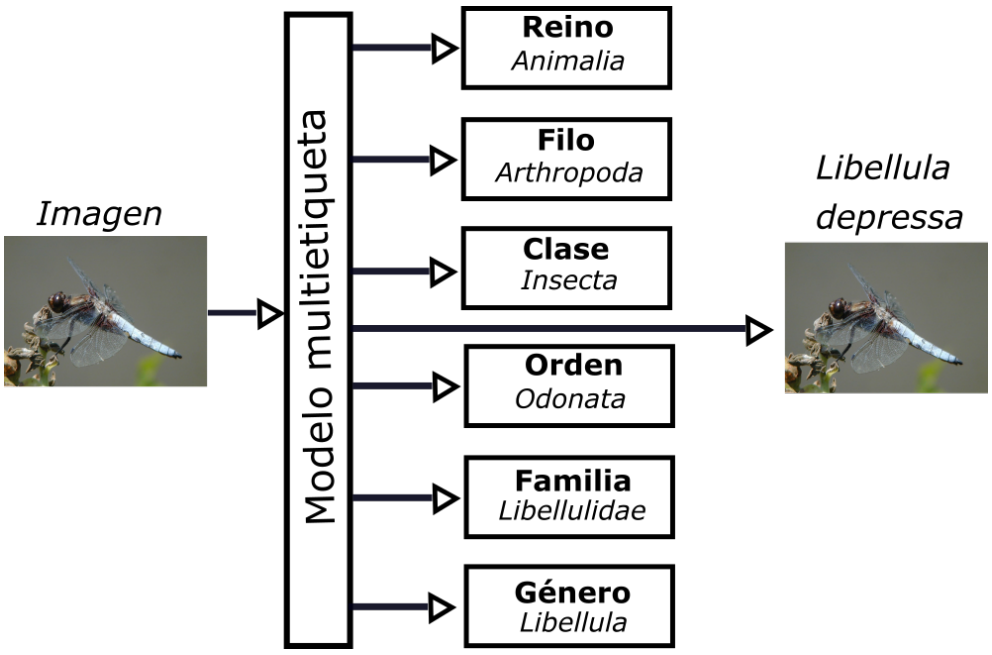


Figura 5.9: Ejemplo de clasificación multietiqueta

5.2.5 Ensemble learning

El *ensemble learning* es la técnica mediante la que se combinan diferentes tipos de clasificadores, con el objetivo de que su combinación mejore los resultados de un único clasificador. Intuitivamente, se puede entender fácilmente, ya que el error de un modelo es subsanado por las decisiones de los demás modelos. La decisión que se toma, es producto lo que deciden todos los clasificadores conjuntamente.

Un punto importante, es que la variedad de datos y las arquitecturas que se utilicen en cada clasificador tienen que diferir lo máximo posible. Ya que si entrenamos exactamente con los mismos datos y con el mismo modelo las decisiones que tomen serán exactamente las mismas. Como en nuestro caso los datos son limitados, lo que se ha propuesto es utilizar un aumentado de datos diferente en cada entrenamiento, cambiando los parámetros que se han usado. A continuación se comentarán las diferentes aproximaciones implementadas.

5.2.5.1 Boosting

La primera aproximación tiene una gran similitud con el Boosting. La idea básica, es que cada red neuronal contribuye a la decisión según el error que tenga en el conjunto de validación. En principio, cuanto mayor sea el número de redes más aumentará el acierto que se obtenga, no obstante a partir de una 4 o 5 redes la mejoría es casi imperceptible.

Algorithm 1: Boosting de un conjunto de CNN

```
PrediccionBoosting (Q, img)
    inputs :
        Q ← Conjunto de clasificadores
        img ← Imagen de entrada
    output:
        Etiqueta de la imagen

    prediccionFinal ← ∅;

    foreach clasificador ∈ Q do
        α ← 1 - error(clasificador)
        prediccion ← α· predecir(clasificador,img)
        prediccionFinal.add(prediccion);

    return argmax(prediccionFinal);
```

Como se puede ver en el algoritmo, por cada clasificador calculamos un valor de confianza, que será mayor cuanto más acierto tenga el modelo. A continuación, multiplicamos ese valor por la predicción que haya hecho de la imagen que queremos identificar. Finalmente, sumamos a la predicción final el resultado del clasificador actual. Este método, aplicado a un conjunto de clasificadores nos permite combinar sus resultados, ponderando con mayor puntación aquellos que tengan un mejor rendimiento.

5.2.5.2 Stacking

El *stacking* consiste en entrenar un algoritmo de aprendizaje que aprenda a combinar las predicciones de otros algoritmos de aprendizaje. El primer paso es entrenar a las distintas arquitecturas con los datos disponibles. A continuación, se entrena el algoritmo combinador para hacer una predicción final usando como entradas todas las predicciones de los otros algoritmos.

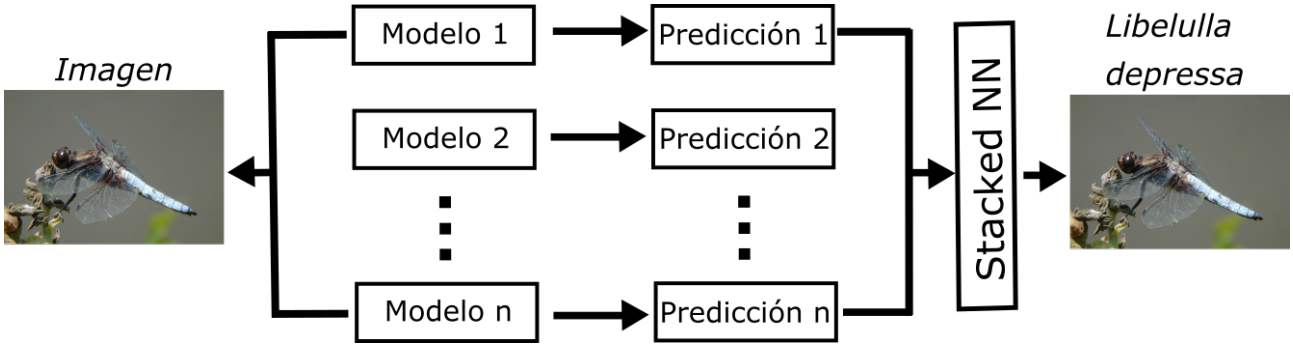


Figura 5.10: Arquitectura stacked

En nuestro caso, utilizamos una fully-connected simple que nos permite combinar los resultados de los distintos modelos. En la Figura 5.10 podemos ver cómo se comporta el Stacking, la imagen de entrada es clasificada independientemente por cada modelo, y a continuación, otra red neuronal es la encargada de combinar estas predicciones y dar un resultado final.

Un punto importante es que los diferentes modelos tienen que haber sido entrenados con parámetros y datos diferentes, de este modo cada clasificador *piensa* de una forma diferente, y la combinación de ellos dará un rendimiento mejor. El objetivo principal de este método es que el meta-algoritmo de aprendizaje puede obtener la combinación óptima de los distintos modelos.

5.2.5.3 Clasificadores especializados

Otra de las estrategias probadas para mejorar el rendimientos de este sistema es entrenar un clasificador para cada filo. El objetivo de este método, es que cada clasificador se especialice en diferenciar especies que tienen más similitud entre sí, reduciendo el espacio de búsqueda que tiene que buscar cada modelo. Para ello, primero necesitamos entrenar un modelo que identifique únicamente el filo de cada imagen, y dependiendo de su predicción, esa imagen se le pase al clasificador correspondiente.

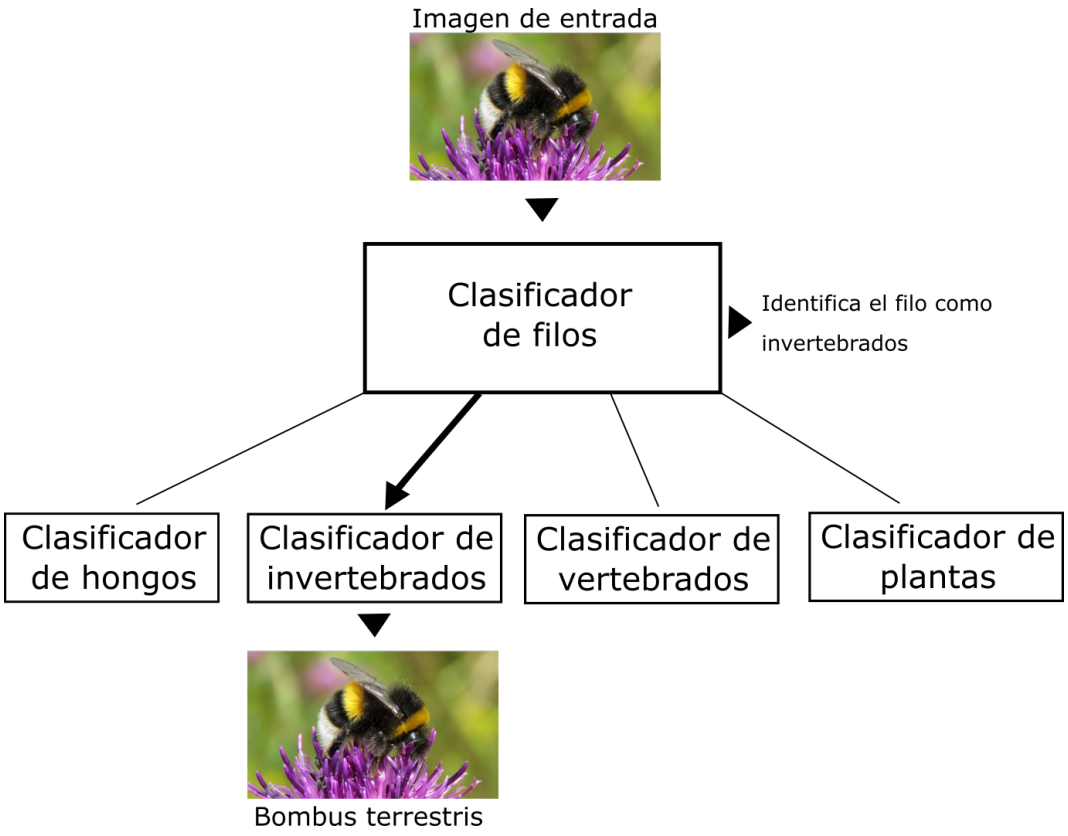


Figura 5.11: Arquitectura con modelos especializados

En el conjunto de datos se incluyen imágenes de cuatro filos diferentes:

- **Basidiomycota:** División del reino Fungi que incluye los hongos que producen basidios con basidiosporas.
- **Tracheophyta:** Abarcan a las plantas vasculares o traqueofitas.
- **Arthropoda:** Incluye animales invertebrados dotados de un esqueleto externo y apéndices articulados, entre otros, insectos, arácnidos, crustáceos y miriápodos.
- **Chordata:** Se incluyen, entre otros las aves y los mamíferos, que pueden elevar y mantener constante la temperatura del cuerpo.

Básicamente, estos cuatro filos se pueden resumir en invertebrados, plantas, hongos y vertebrados. Para cada uno de ellos se deberá construir un modelo especializado. En la Figura 5.11 se ve un ejemplo de funcionamiento de este sistema, se recibe una imagen de entrada y el clasificador de filos la identifica como invertebrado, por que se le pasa a ese clasificador, que nos da la especie pertinente.

5.2.6 Clasificación por niveles

El hecho de que los imágenes estén organizadas por categorías taxonómicas nos puede servir para mejorar el rendimiento del modelo, la idea básica es predecir cada una de las etiquetas de cada nivel, y a continuación, intentar ver cuál es el camino más consistente en el árbol que se genera. Esta técnica se diferencia de los clasificadores especializados, ya que aquí usamos un único clasificador multietiqueta. La forma en la que recorramos el árbol es determinante para realizar la predicción, las dos técnicas que se proponen son las siguientes:

- **Elegir la categoría con mayor probabilidad en cada nivel**

  Esta técnica consiste en ir seleccionando en cada nivel la categoría taxonómica que predice el clasificador, el recorrido se realizaría desde las categorías más generales hacia las más específicas, con el objetivo de ir descartando ramas del árbol. En la Figura 5.12 se puede ver de forma esquemática el funcionamiento de este sistema.

- **Sumar las probabilidades de todas las categorías**

  En este caso se propone propagar hacia abajo las probabilidades de cada categoría taxonómica, a cada hijo se le suma la probabilidad que se haya obtenido en su categoría padre, y este a su vez lo transmite a su hijo. La predicción final será aquella predicción que tenga un mayor valor en el último nivel.

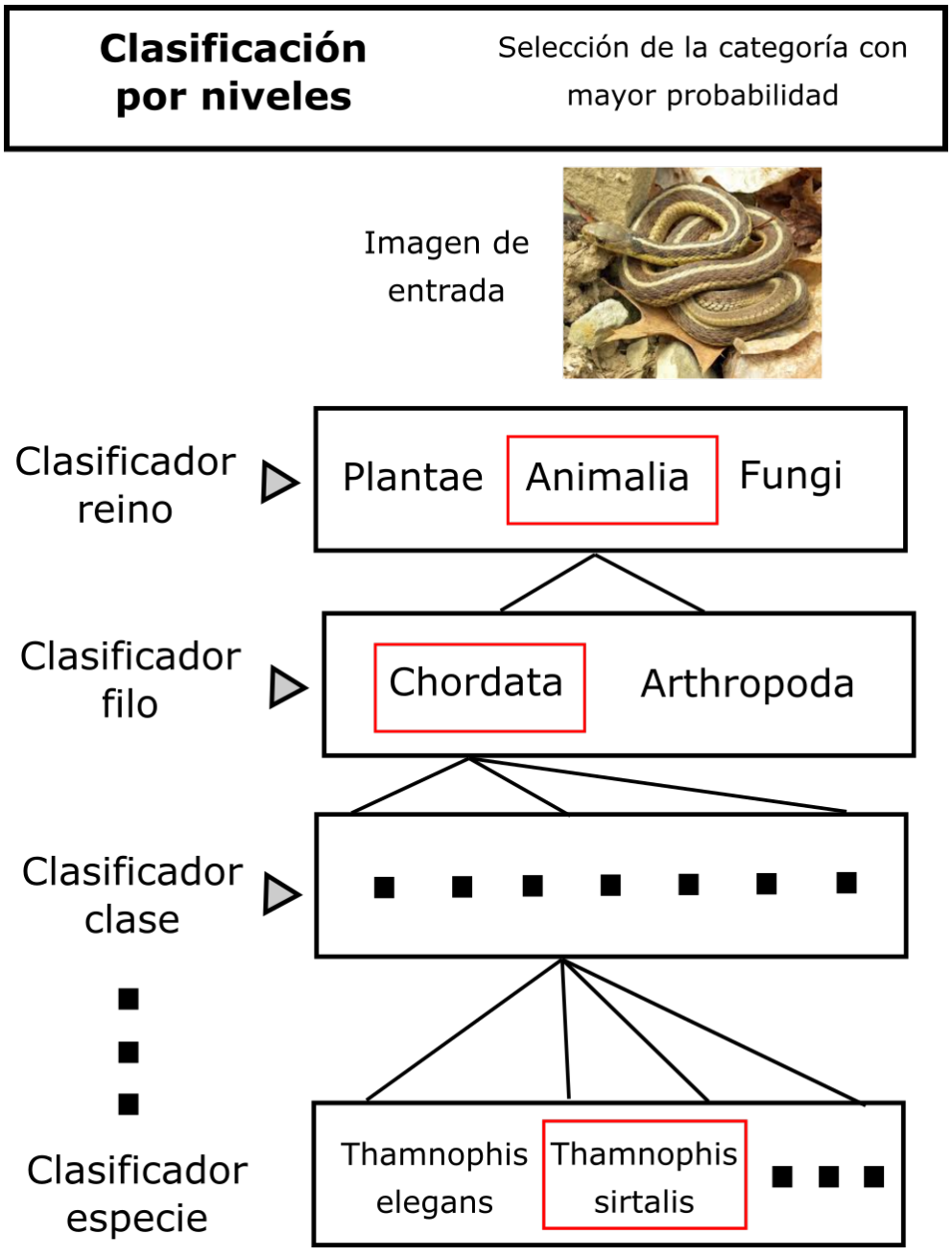


Figura 5.12: Clasificación de niveles

6 Experimentación

6.1 Resultados de los experimentos

En este punto se van a analizar y mostrar los resultados de los diversos experimentos realizados durante el desarrollo del proyecto. Por cada caso, se explicarán los parámetros que se han utilizado. En casi todos los experimentos, los resultados se han obtenido utilizando un 70% de los datos disponibles de iNaturalist para el entrenamiento, y dejando el 30% restante para la validación del modelo.

6.1.1 Primer experimento - Resnet50

En la figura 6.1 aparecen los resultados del primer experimento con una Resnet50. En esta prueba no se ha utilizado ninguna mejora, ni siquiera se han inicializado los pesos con Imagenet. Como se puede observar en la gráfica, la diferencia de acierto entre el entrenamiento y el test es muy alta.

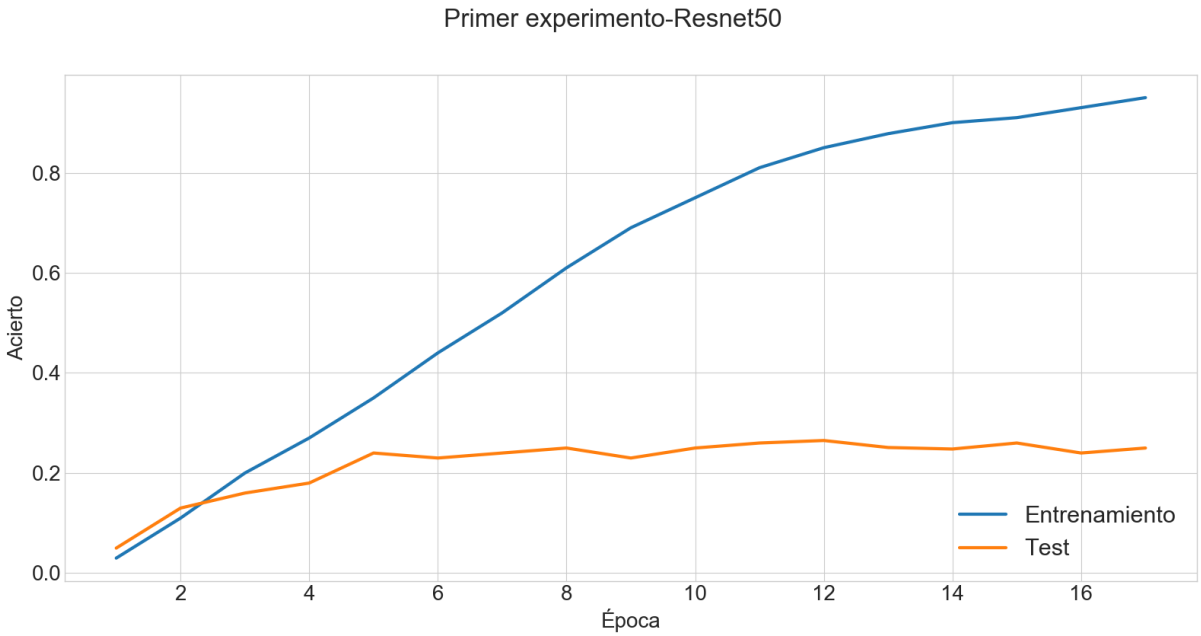


Figura 6.1: Acierto en el género Bombus de Biodiversidad

Modelo	Resolución	Aumentado de datos	Imagenet
ResNet50	250x250	No	No

Tabla 6.1: Parámetros en el primer experimento

El primer experimento que hemos realizado ha sido el que peor resultado ha dado, esto se debe a que no se ha aplicado ninguna mejora de ningún tipo, simplemente se le pasan los datos sin equilibrar, y se inicializa la red con pesos aleatorios. La resolución es también más baja que la que se ha utilizado en experimentos positivos. También se usa la ResNet50, que la red que peor resultado nos ha dado.

6.1.2 Aumentado de datos

La primera experimentación que se ha realizado es entrenar al modelo simplemente con las imágenes del conjunto de datos, sin aplicar ninguna técnica o transformación a las imágenes ni corregir el desbalanceo de datos. En el segundo experimento se ha utilizado *Data augmentation*, aplicándolo de la forma que ha sido expuesta en apartados anteriores. Los parámetros que se han utilizado ambas pruebas se muestran en la Tabla 6.2.

Modelo	Resolución	Aumentado de datos	Taxonomía
ResNet50	250x250	No	No
ResNet50	250x250	Si	No

Tabla 6.2: Parámetros usados en el experimento de data augmentation

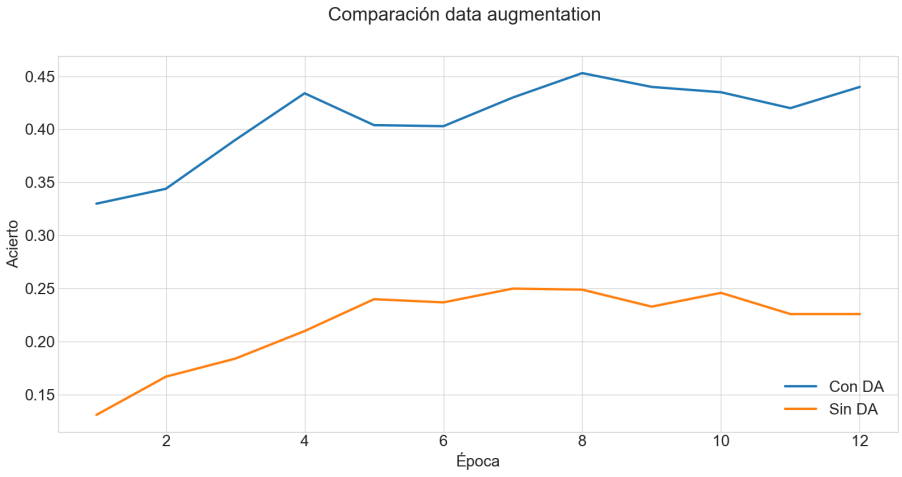


Figura 6.2: Resultados de utilizar data augmentation

La Figura 6.2 muestra los resultados de ambos experimentos sobre el test de validación, el acierto medido es el TOP-1. Como se puede observar, el rendimiento del modelo que no

aplica aumentado de datos es muy inferior al que sí que lo aplica, con tan solo un 27% de acierto. El gran desbalanceo de los datos provoca que el modelo no generalice bien, y tiende a clasificar las imágenes con las clases más representadas.

Recordemos que el balanceo de los datos es un cuestión clave en los sistemas de aprendizaje automático supervisados, ya que los algoritmos de aprendizaje tienden a favorecer las categorías con más elementos. La generación de datos sintéticos tiene como objetivo corregir este desbalanceo, y como vemos en la gráfica, añadiendo esta técnica se obtiene una gran mejora en el acierto, que pasa de un 27% a un 45%.

6.1.3 Resolución

La resolución es uno de los factores claves en este problema, ya que cuanto mayor sea, con más detalle podrá nuestro modelo apreciar las sutiles diferencias que tiene cada especie. Para comprobar hasta que punto influye la resolución en el rendimiento final, se han llevado a cabo varios experimentos con diferentes resoluciones. En las tablas 6.3 y 6.4 se muestran los parámetros usados en cada caso.

Modelo	Resolución	Aumentado de datos	Taxonomía
ResNet50	250x250	No	No
ResNet50	300x300	No	No

Tabla 6.3: Parámetros usados en el experimento 1 de resolución

Modelo	Resolución	Aumentado de datos	Taxonomía
DenseNet201	250x250	Si(Sin crop)	No
DenseNet201	300x300	Si(Sin crop)	No

Tabla 6.4: Parámetros usados en el experimento 2 de resolución

En las Figuras 6.3 y 6.4 se puede observar la gran mejoría que aporta aumentar la resolución de las imágenes. En el primer experimento, el incremento en el acierto es del 9%, mientras que en el segundo es del 6%. En este caso, hemos pasado de 250x250 a 300x300 píxeles, pero si aumentáramos aún más la resolución, probablemente también lo haría el acierto. No obstante, 300x300 píxeles es la máxima resolución que se ha podido entrenar manteniendo un tamaño de batch aceptable.

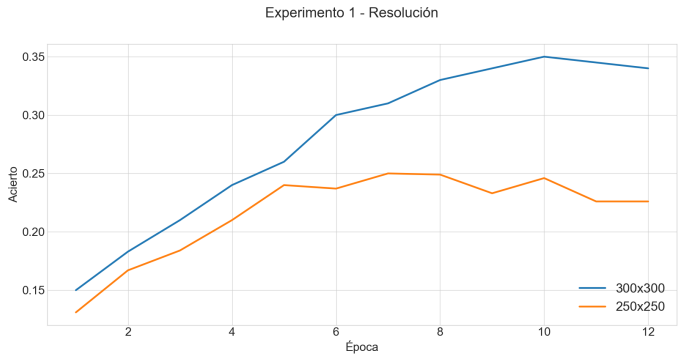


Figura 6.3: Experimento resolución 1

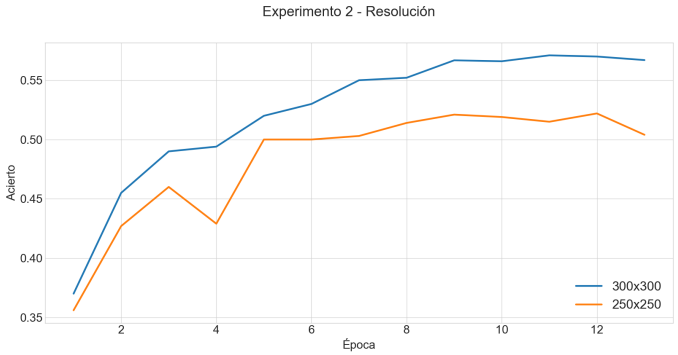


Figura 6.4: Experimento resolución 2

6.1.4 Arquitecturas

Durante el desarrollo se han utilizado varias arquitecturas, concretamente las cinco que se han utilizado han sido la *DenseNet201*, la *ResNet50*,la *InceptionV1*, *EfficientNetB3* y la *EfficientNetB5*.

Modelo	Resolución	Aumentado de datos	Taxonomía
DenseNet201	250x250	Si(Sin crop)	No
ResNet50	250x250	Si(Sin crop)	No
InceptionV1	250x250	Si(Sin crop)	No
EfficientNetB3	300x300	Si(Sin crop)	Si
EfficientNetB5	300x300	Si(Sin crop)	Si

Tabla 6.5: Parámetros usados en el experimento 2 de resolución

En las Figuras 6.5 y en 6.6 aparecen los resultados en el Top 1 y el Top 5 de las diferentes arquitecturas. La EfficientNetB3 es la que mejor rendimiento ha dado en las pruebas. La peor ha sido la Resnet50.

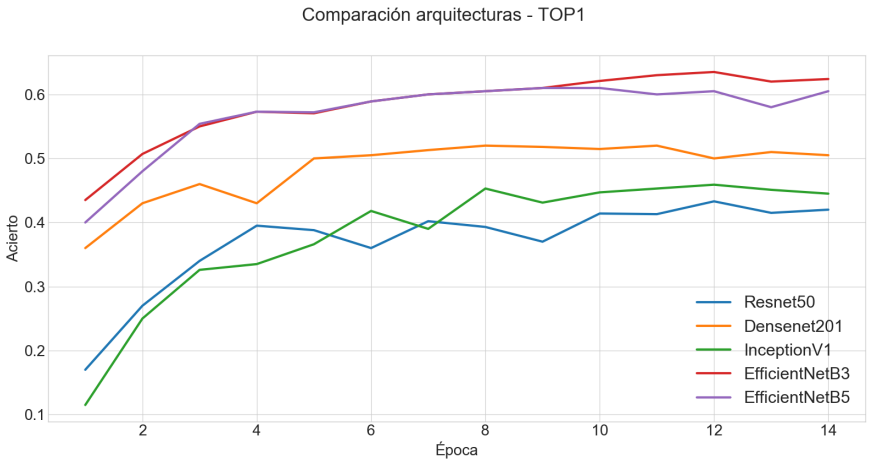


Figura 6.5: Acierto Top 1 - Experimento con diferentes arquitecturas

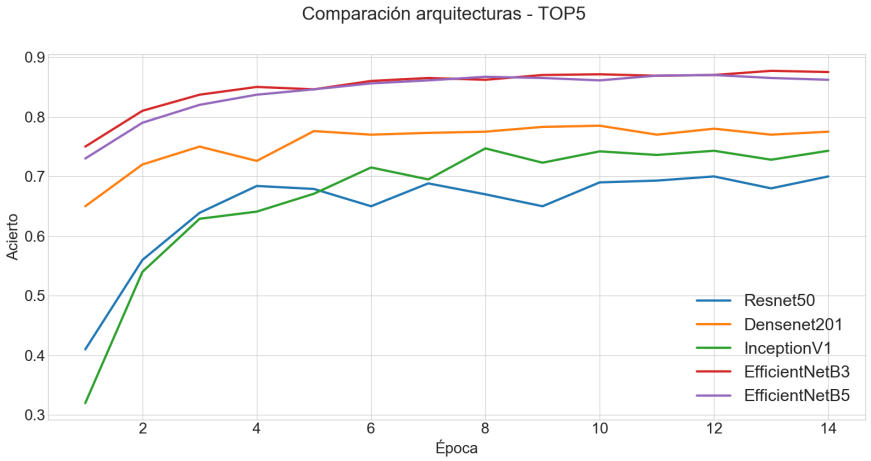


Figura 6.6: Acierto Top 5 - Experimento con diferentes arquitecturas

6.1.5 Crop

Uno de los experimentos que se ha llevado a cabo está relacionado con el *crop* o el recorte de las imágenes. La comparación que se ha realizado ha sido con un entrenamiento sin *crop*, otro con un *crop* normal, y otro piramidal como se ha explicado en apartados anteriores. Los parámetros que se han usado se muestran en la tabla.

Como se observa en la Figura 6.7 la mejora del *crop* es bastante ligera (un poco mayor en el piramidal), mejora aproximadamente un 2 % respecto al caso en el que no se utiliza. Aunque la mejora sea ligera, los resultados nos indican que su uso es recomendable.

Modelo	Resolución	Aumentado de datos	Taxonomía
DenseNet201	300x300	Si(Sin crop)	Si
DenseNet201	300x300	Si(Con crop)	Si
DenseNet201	300x300	Si(Crop triangular)	Si

Tabla 6.6: Parámetros usados en el experimento del crop

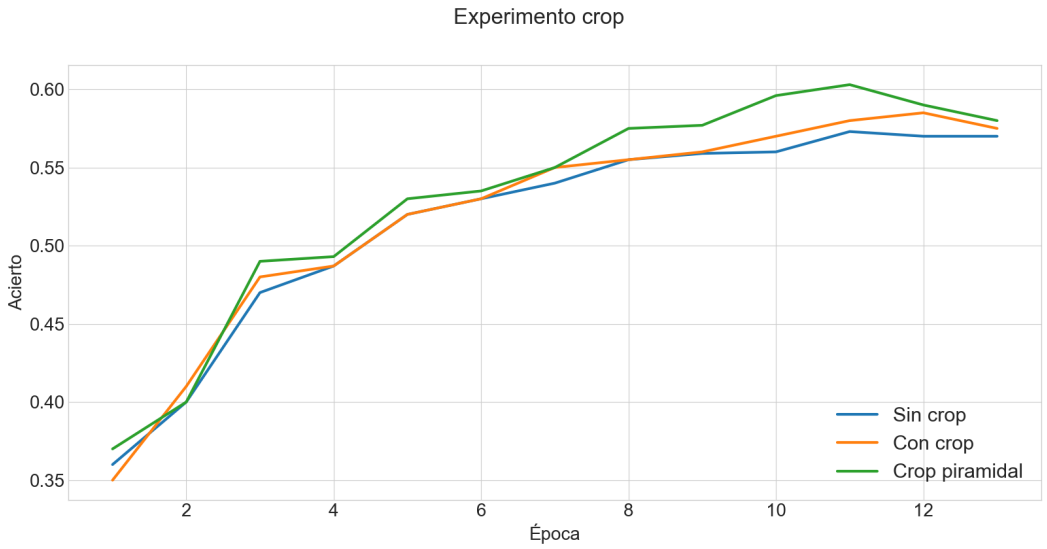


Figura 6.7: Acierto - Experimento con crop

6.1.6 Uso de datos de años anteriores

Para aumentar el tamaño del conjunto de datos y mejorar el rendimiento del sistema, se han utilizado también imágenes de competiciones de *iNaturalist* de años anteriores. Algunas de las especies con las que estamos trabajando ya se utilizaron en años anteriores, no obstante, la mayor parte de estos datos pertenecen a las categorías más comunes, por lo que ya tenemos datos suficientes y no tiene sentido aumentar el desbalanceo en el conjunto de datos. Para el experimento, solamente se han seleccionado aquellas imágenes que pertenecen a especies infrarepresentadas, con un total de 10.000 imágenes añadidas.

Modelo	Resolución	Aumentado de datos	Taxonomía	Datos de años anteriores
DenseNet201	300x300	Si(Con crop)	Si	No
DenseNet201	300x300	Si(Con crop)	Si	Si

Tabla 6.7: Parámetros usados en el experimento con datos de años anteriores

En la Figura 6.8 se compara el uso de datos de años anteriores. Como se puede observar, hay una ligera mejora aumentado la cantidad de datos, de aproximadamente del 2%. Si tuviéramos más imágenes de las clases infrarrepresentadas, el rendimiento probablemente

aumentaría más.

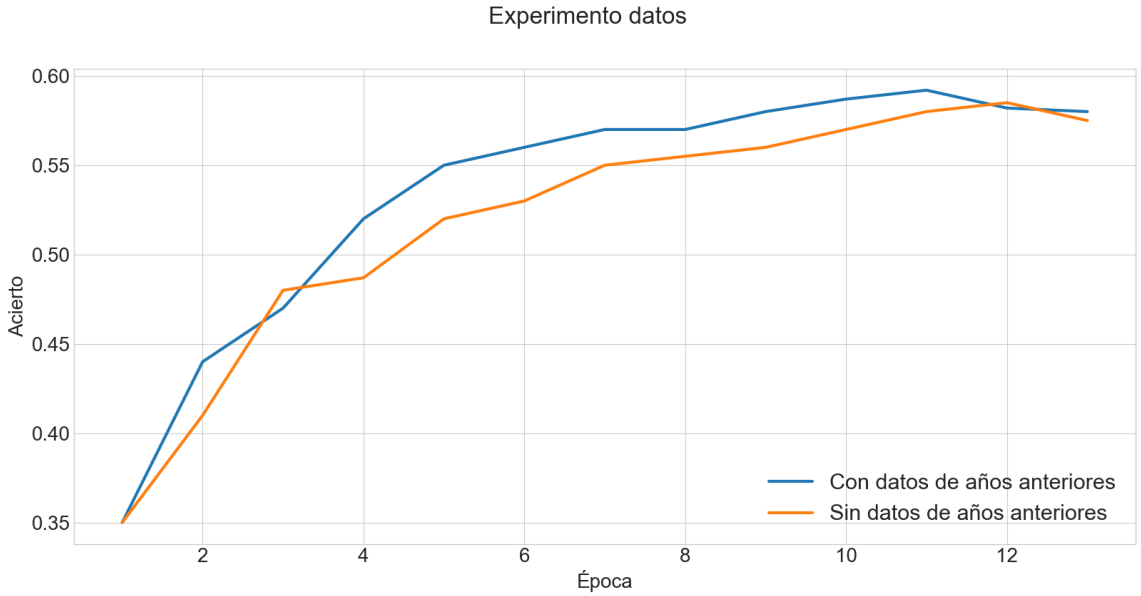


Figura 6.8: Acierto - Experimento con datos de otros años

6.1.7 Comparación entre entrenamiento y test

En esta sección vamos a realizar una comparación entre los resultados obtenidos entre el conjunto de test y el del entrenamiento. El modelo CNN que se ha seleccionado para el experimento es el que mejor resultado nos ha dado en todas las pruebas. Se trata del EfficientNetB3. En la Tabla 6.8 aparecen los resultados de acierto obtenidos para cada categoría taxonómica.

Categoría	Acierto test	Acierto entrenamiento
Reino	0.99	0.999
Filo	0.98	0.99
Clase	0.97	0.99
Orden	0.91	0.98
Familia	0.89	0.97
Género	0.88	0.97
Especie	0.63	0.95

Tabla 6.8: Resultados de las categorías taxonómicas

La Figura 6.9 muestra una comparativa entre los resultados obtenidos en el conjunto de entrenamiento y el de test. Mientras que en el entrenamiento el acierto va creciendo hasta casi un 100%, en el test llega un momento que se estanca y se queda en un 63% de acierto.

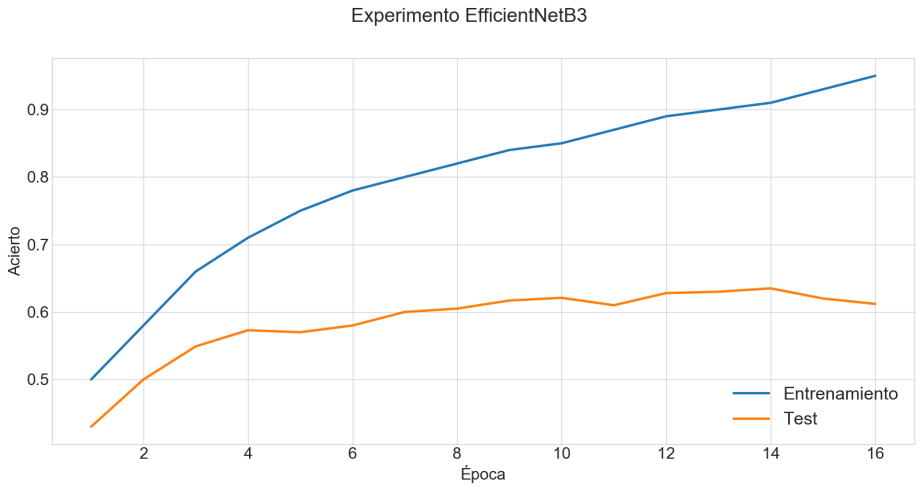


Figura 6.9: Comparación entre entrenamiento y test

6.1.8 Resumen de las arquitecturas CNN

Modelo	Resolución	Aumentado de datos	Taxonomía	TOP 1	TOP 5
ResNet50	250x250	No	No	0.27	0.62
ResNet50	300x300	No	No	0.35	0.65
ResNet50	250x250	Si (Sin crop)	No	0.45	0.73
InceptionV1	250x250	No	No	0.34	0.63
InceptionV1	250x250	Si (Sin crop)	No	0.46	0.75
DenseNet201	250x250	Si (Sin crop)	No	0.51	0.77
DenseNet201	250x250	Si (Sin crop)	Si	0.52	0.79
DenseNet201	250x250	Si (Con crop)	Si	0.53	0.80
DenseNet201	300x300	Si (Sin crop)	Si	0.58	0.83
DenseNet201	300x300	Si (Con crop)	Si	0.59	0.84
DenseNet201	300x300	Si (Crop piramidal)	Si	0.61	0.85
EfficientNetB5	300x300	Si (Crop piramidal)	Si	0.62	0.88
EfficientNetB3	300x300	Si (Crop piramidal)	Si	0.63	0.89

Tabla 6.9: Tabla resumen con los resultados de las diferentes arquitecturas

En la Tabla 6.9 se muestra de forma resumida los resultados de las diferentes arquitecturas CNN probadas. Los modelos que tienen taxonomía son las que realizan la clasificación multietiqueta de las diferentes categorías taxonómicas.

La arquitectura CNN que mejor resultado ha dado ha sido la EfficientNetB3, utilizando una resolución de 300x300 píxeles y usando aumentado de datos con un *crop* piramidal. El acierto de este modelo es 63% en el último nivel, el mejor obtenido sin usar las técnicas de *Ensemble learning*, cuyos resultados mostraremos a continuación.

6.1.9 Ensemble Boosting

En este punto vamos a mostrar los resultados que hemos obtenido con la aplicación del Ensemble Boosting, como hemos explicado anteriormente, este sistema trata de combinar los resultados de varias redes neuronales, para ello el voto de cada red se pondera según su porcentaje de acierto. Por cada ensemble vamos a mostrar sus parámetros.

Ensemble 1

Modelo	Resolución	Aumentado de datos
DenseNet201	250x250	Si
ResNet50	250x250	Si
InceptionV1	250x250	Si

Tabla 6.10: Párametros Ensemble 1

Ensemble 2

Modelo	Resolución	Aumentado de datos
DenseNet201	300x300	Si
DenseNet201	300x300	Si(Diferente DA)
InceptionV1	300x300	Si

Tabla 6.11: Párametros Ensemble 2

Ensemble 3

Modelo	Resolución	Aumentado de datos
DenseNet201	300x300	Si
EfficientNetB3	300x300	Si(Diferente DA)
EfficientNetB5	300x300	Si

Tabla 6.12: Párametros Ensemble 3

Ensemble	Acierto - TOP1	Acierto - TOP5	Acierto Kaggle
Ensemble 1	0.59	0.88	0.57
Ensemble 2	0.66	0.93	0.64
Ensemble 3	0.71	0.94	0.67

Tabla 6.13: Resultados Ensemble

En las Tablas 6.10, 6.11 y 6.12 aparecen los parámetros utilizados en cada ensemble. En la Tabla 6.13 aparecen los resultados obtenidos en cada ensemble. Como podemos observar, este método nos da una significativa mejora de aproximadamente 5% respecto a utilizar un único modelo.

El *Acierto Kaggle* nos indica la puntación que se ha obtenido en la competición, que es un poco menor que la obtenida en nuestras mediciones. El mejor resultado que se ha obtenido con cualquier método ha sido con el *Ensemble 3*, con el que se ha conseguido un acierto del 71% en la clasificación de especies del conjunto iNaturalist.

6.1.10 Clasificación por niveles

La clasificación por niveles consiste en ir recorriendo el árbol desde las categorías taxonómicas más generales hasta las más específicas. En la Tabla 6.14 se muestra el modelo utilizado para realizar el experimento.

Modelo	Resolución	Aumentado de datos
DenseNet201	250x250	Si

Tabla 6.14: Párametros clasificación por niveles

La Tabla 6.15 contiene los resultados de la experimentación, como se puede observar la clasificación por niveles no nos aporta ninguna mejora sobre el modelo CNN. Esto se debe principalmente a que el sistema tiene problemas en identificar la especie en el último nivel, pero no en el camino hasta ahí. Pongamos como ejemplo la imagen de una culebra, nuestro modelo es muy efectivo para identificar que en la foto aparece este animal, sin embargo, el problema está en distinguir entre las más de 20 especies de culebras diferentes que aparecen en el conjunto de datos. Por ello, esta técnica se ha descartado al no obtener resultados positivos.

Acierto CNN	Escoger categoría mayor probabilidad	Sumar todas las probabilidades
0.52	0.51	0.52

Tabla 6.15: Resultados clasificación por niveles

6.1.11 Clasificación de filos

Esta técnica, como se ha explicado en apartados anteriores, consiste en entrenar un clasificador especializado en cada uno de los 4 filos que existen en iNaturalist. Para realizar una primera prueba, se ha seleccionado el filo Tracheophyta, que abarca a las plantas vasculares. El procedimiento para ver si este método nos aporta alguna mejora es comparar el acierto que tiene un modelo entrenado con todas las especies y otro especializado únicamente en plantas.

Modelo	Resolución	Aumentado de datos
DenseNet201	300x300	Si

Tabla 6.16: Párametros clasificación por filos

En la Tabla 6.17 se muestran los resultado de esta técnica. La mejora que nos otorga utilizar clasificadores especializados es muy leve. La razón detrás de esta poca mejoría es la misma que en la clasificación por niveles, el sistema sigue teniendo problemas en identificar la especie en el último nivel, por lo que especializarse en un filo no le da una mejoría muy notoria. Al no obtener grandes resultados, este sistema se ha descartado.

Acierto en el filo Tracheophyta

CNN con todas las especies	CCN especializado
0.61	0.615

Tabla 6.17: Resultados clasificación por filos

6.2 Clasificación de vídeo

Uno de las aplicaciones del clasificador más interesantes es la identificación de vídeo. Para realizar esta prueba, se han conseguido fragmentos de vídeo en los que aparecen una de las más de mil especies que hemos entrenado. En las imágenes de entrenamiento, se combinan imágenes de alta calidad con otras en las que el elemento a identificar apenas aparece en la imagen. En este caso, se ha intentado que todos los fragmentos tengan una resolución alta y el elemento aparezca de forma clara. Para realizar la clasificación, el modelo identifica 10 fotogramas seguidos, y calcula la suma de las probabilidades de todos los fotogramas. Este método se va aplicando durante todo el vídeo.

En el siguiente enlace aparece un vídeo que muestra este tipo de clasificación, a la izquierda aparece la etiqueta correcta, y a la derecha el vídeo con la etiqueta que el sistema está prediciendo:

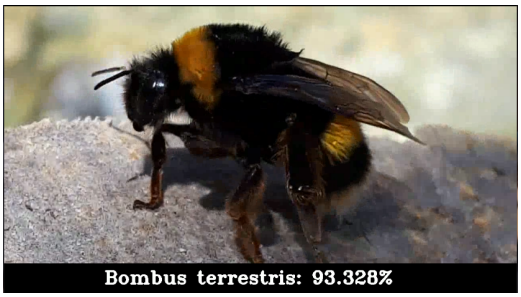
`https://www.youtube.com/watch?v=rzEIYt4Gj0A&t=6s`

A continuación mostraremos los resultados de la clasificación aplicada a diferentes vídeos, abajo de cada fotograma aparece la especie que el modelo ha identificado, el valor de la derecha indica lo seguro que está de la imagen pertenezca a esa especie:

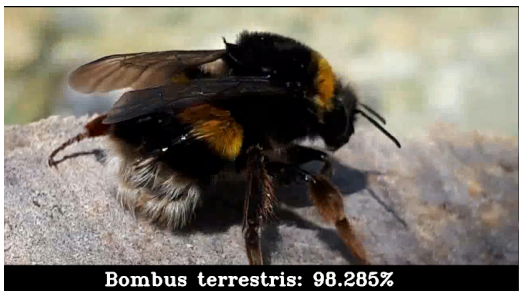
(a) Fragmento 1

(b) Fragmento 2

Figura 6.10: Clasificación en vídeo de *Bombus vosnesenskii*



(a) Fragmento 1

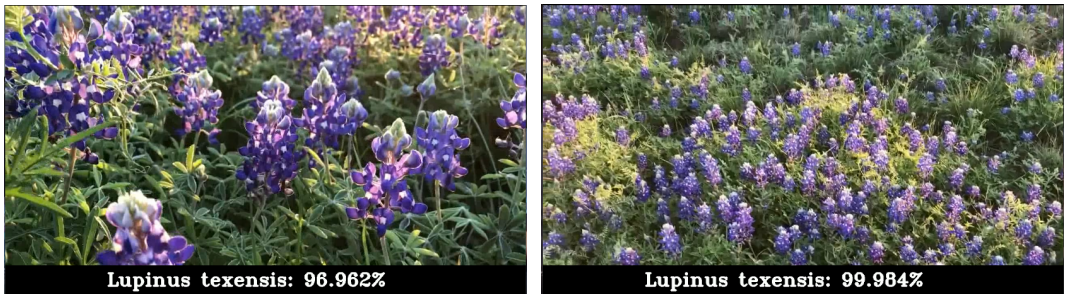


(b) Fragmento 3

Figura 6.11: Clasificación en vídeo de *Bombus terrestis*

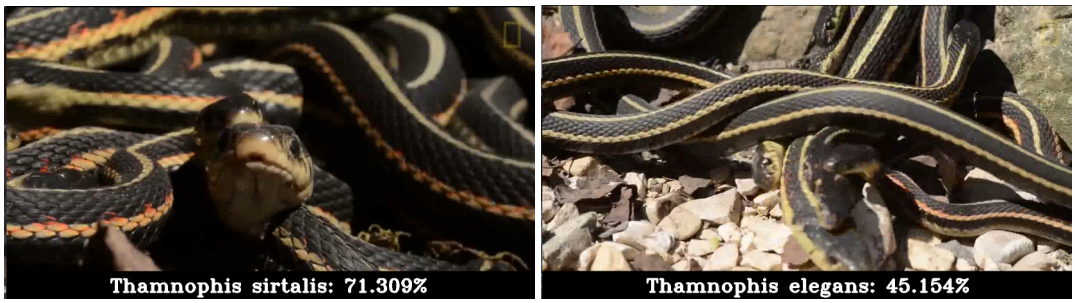
En la Figura 6.10 aparece la clasificación sobre una abeja, perteneciente concretamente a la especie *Bombus vosnesenskii*, la identificación se realiza de forma correcta para todos los fotogramas del vídeo. En este caso es muy interesante realizar una comparación con la Figura 6.11, ya que podemos ver cómo el sistema es capaz de distinguir que especie de abeja exactamente, tanto en situaciones donde el insecto está de perfil como cuando esta de espaldas. En el caso de las abejas, hay un total de 20 especies diferentes, todas ellas con un grado de similitud visual bastante alto. Que el sistema pueda identificarlas correctamente en distintas posiciones es una muestra de la robustez del modelo.

En la Figura 6.12 aparecen fragmentos del vídeo de una planta, en este caso la especie a clasificar es *Lupinus texensis*. En el vídeo aparecen primeros planos de la planta y otros en los está mas alejado. En todos los casos el sistema acierta la especie con mucha seguridad, cercana al 100%.



(a) Fragmento 1 **(b)** Fragmento 2

Figura 6.12: Clasificación en vídeo de *Lupinus texensis*



(a) Fragmento 1 **(b)** Fragmento 2

Figura 6.13: Clasificación en vídeo de *Thamnophis sirtalis*

La Figura 6.13 muestra los fragmentos de un vídeo especialmente interesante por varios motivos. El primer motivo es que las culebras que aparecen se mueven muy rápido, por lo que es una buena forma de comprobar si el sistema funciona bien en imágenes en movimiento, donde algunos fotogramas puede estar borrosos.

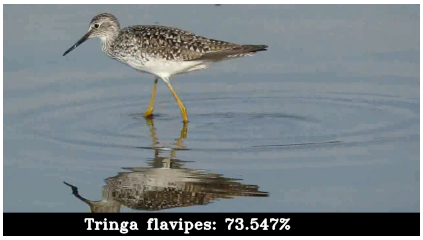
Otro de los factores a analizar, es que en el tercer fragmento el sistema se confunde de especie, ya que la identifica como *Thamnophis elegans*. Esta confusión tiene una explicación muy sencilla, ya que las dos especies son prácticamente idénticas en lo visual. La única diferencia visual es que el *Thamnophis elegans* tiene 8 escamas labiales superiores, mientras que la *Thamnophis sirtalis* solo 7. En la Figura 6.14 se ve claramente las similitudes visuales de ambas especies, que son casi indistinguibles visualmente, por lo que la confusión entre ambas especies por parte del sistema es bastante lógica.

En las Figuras 6.15 y 6.16 se realiza la identificación de dos especies de aves diferentes, y en ambos casos la clasificación es correcta. En estos caso por ejemplo, es muy difícil diferenciar entre especies de ave del mismo género, ya que las diferencias son mínimas.

(a) Thamnophis sirtalis

(b) Thamnophis elegans

Figura 6.14: Especies de culebras similares



(a) Fragmento 1

(b) Fragmento 2

Figura 6.15: Clasificación en vídeo de *Tringa flavipes*

6.3 Biodiversidad Virtual

Biodiversidad Virtual es una plataforma científica y divulgativa basada en el trabajo cooperativo y la participación ciudadana. Consiste en doce galerías temáticas de fotografías digitales geolocalizadas que conforman una base de datos ordenada taxonómicamente. La mayor parte de la flora y fauna que aparece en la web pertenece a especies fotografiadas en la Península Ibérica. Entre las funciones y objetivos de esta organización se encuentran fomentar el estudio de la naturaleza y sus procesos, creando una conciencia en toda la sociedad sobre la conservación y el conocimiento del entorno. En nuestro caso, el objetivo de utilizar esta plataforma para obtener imágenes de algunas categorías taxonómicas que coinciden con las entrenadas para validar nuestro modelo. Muchas de las especies que aparezcan no habrán sido entrenadas en nuestro modelo, no obstante, el propósito es ver si aún así es capaz de acertar alguna categoría taxonómica.

Para realizar este experimento, se ha seleccionado el género *Bombus*, a este orden pertenecen diferentes tipos de abejorroros. En el conjunto de iNaturalist hay un total de 20 especies de abejorros diferentes, mientras que la variedad en Biodiversidad Natural es mayor, ya que hay un total de 40 especies de abejorros, todos ellos han sido fotografiados en la Península Ibérica.

(a) Fragmento 1

(b) Fragmento 2

Figura 6.16: Clasificación en vídeo de *Setophaga petechia*

(a) Especie 1 de Bombus

(b) Especie 2 de Bombus

Figura 6.17: Imágenes de Bombus de Biodiversidad Virtual

En la Figura 6.17 se ven dos imágenes de Bombus extraídas de Biodiversidad Virtual, en la mayoría de las fotografías el elemento a identificar aparece de forma clara, y no aparece alejado. Esta característica es muy útil para una posible mejora en el rendimiento del modelo.

Orden	Cantidad de imágenes	Especies
Bombus	1802	39

Tabla 6.18: Género Bombus en Biodiversidad Virtual

En la Figura 6.18 vemos el acierto que ha tenido nuestro modelo para acertar el género de las imágenes de Biodiversidad Virtual. Ha acertado en un total de 1674 fotografías, lo que implica un 93% de acierto en el orden. Un 5% más que en el conjunto de iNaturalist. Estos resultados son muy positivos, ya que la mayoría de las especies que aparecen en las fotografías no las ha visto nunca el modelo, no obstante sí que es capaz de identificar de forma correcta su categoría taxonómica más cercana, que en este caso es el género.

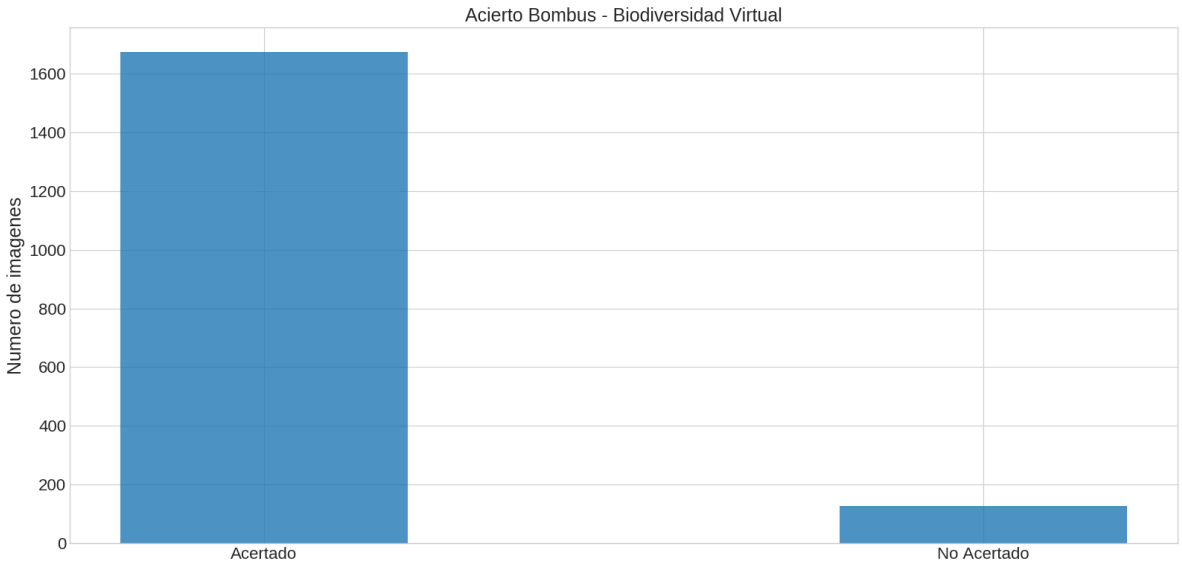


Figura 6.18: Acierto en el género Bombus de Biodiversidad

7 Conclusiones

Como conclusión de este TFG, se va a hacer un repaso punto por punto de todo lo conseguido. Además, también se incluyen algunas reflexiones sobre lo aprendido a lo largo del proyecto.

Estos son los resultado obtenidos expuestos de forma sintética:

- Se ha conseguido procesar y tratar los diferentes datasets de forma efectiva. La cantidad de datos necesarios para el entrenamiento ocupaban un total de 200GB, por lo que realizar un manejo eficiente de esta información ha sido clave en el desarrollo del proyecto. Además, se han fusionado datos de competiciones de iNaturalist en años anteriores.

- Se han utilizado con éxito diferentes técnicas de *Data Augmentation*. Debido al gran desbalanceo entre las diferentes categorías, el DA ha mejorado notablemente el rendimiento de nuestro modelo.

- Se han probado múltiples parámetros y arquitecturas durante la experimentación, con el objetivo de compararlas y mejorar el acierto de nuestro sistema.

- La cantidad de técnicas y algoritmos usados durante la experimentación ha sido realmente amplia. Entre estas técnicas encontramos el uso de ensembles, el stacked learning, la clasificación por niveles taxónomica y los clasificadores especializados en filos.

- Se ha conseguido aplicar nuestro sistema a vídeos reales donde aparecen especies en movimiento. Este punto ha sido uno de los más satisfactorios del proyecto, ya que nos ha permitido ver en acción nuestro modelo en entornos reales, además, los resultados obtenidos han sido muy positivos, siendo capaz de distinguir entre más de mil especies con una alta robustez.

- Las imágenes de Biodiversidad Virtual nos han permitido probar con éxito nuestro sistema en imágenes que contienen especies para las que no ha sido entrenado explícitamente.

Uno de los puntos que podría mejorarse es el resultado obtenido en la competición de Kaggle, que aunque no es malo, podría ser mejor. No obstante, hay que puntualizar estos resultados, ya que este tipo de competiciones están enfocadas en conseguir maximizar la puntuación para el reto, por lo que en muchas ocasiones que alguien tenga una mejor puntuación no implica necesariamente que su modelo funcione mejor en un entorno real, sino que se ha lanzado numerosos envíos al sistema para intentar predecir las etiquetas correctas. Además,

también hay que tener en cuenta que muchos de los equipos que participan pertenecen a grandes compañías tecnológicas como Google o Facebook, con unos conocimientos técnicos y unos recursos computacionales mucho mayores.

A pesar de esto, considero que el resultado del trabajo ha sido muy satisfactorio, ya que se ha conseguido construir un modelo con un alto rendimiento que resuelve el problema planteado, y que utiliza los últimos avances en inteligencia artificial para mejorar la identificación de la biodiversidad en nuestro planeta.

Bibliografía

[1] G. OU and Y. Murphey, "Multi-class pattern classification using neural networks," *Pattern Recognition*, vol. 40, no. 1, pp. 4–18, 2007.

[2] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, pp. 415–425, 2002.

[3] E. Allwein, R. Schapire, and Y. Singer, "Reducing multiclass to binary: A unifying approach for margin classifiers," *Journal of machine learning research*, vol. 1, pp. 113–141, 2000.

[4] R. Babbar, I. Partalas, E. Gaussier, and M.-R. Amini, "On flat versus hierarchical classification in large-scale taxonomies," *27th Annual Conference on Neurla INformation Processing Systems (NIPS 26)*, pp. 1824–1832, 2013.

[5] Y. Taigman, M. R. M. Yang, and L. Wolf., "Deepface: Closing the gap to human-level performance in face verification," *In CVPR*, 2014.

[6] F. Schroff, D. Kalenichenko, and J. P. Facenet, "A unified embedding for face recognition and clustering," *In CVPR*, 2015.

[7] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," *In BMVC*, 2015.

[8] S. Cui, Y. Song, Y. Sun, C. Howard, and A. Belongie, "Large scale fina-grained categorization and domain-specific transfer learning," *In CVPR*, 2018.

[9] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-ucsd birds," 2010.

[10] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie., "The caltech-ucsd birds-200-2011 dataset," 2011.

[11] T. Berg, J. Liu, S. W. Lee, M. L. Alexander, D. W. Jacobs, and P. N. Belhumeur, "Birdsnap: Large-scale fine-grained visual categorization of birds," *In CVPR*, 2014.

[12] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Feir, "Novel dataset for fine-grained image categorization," *FGVC Workshop at CVPR*, 2011.

[13] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar., "Cats and dogs dataset," *In CVPR*, 2012.

[14] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, and A. Khosla, "Imagenet large scale visual recognition challenge," *In IJCV*, 2015.

[15] Y. S. Y. C. C. S. A. S. H. A. P. P. S. B. Grant Van Horn, Oisin Mac Aodha, "The inaturalist species classification and detection dataset," *The Quarterly Journal of Experimental Psychology*, vol. 18, no. 4, pp. 362–365, 2018.